Jungermann, Felix; Morik, Katharina

Working Paper

Enhanced services for targeted information retrieval by event extraction and data mining

Technical Report // Sonderforschungsbereich 475, Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, No. 2008,04

Provided in cooperation with:
Technische Universität Dortmund

Enhanced Services for Targeted Information Retrieval by Event Extraction and Data Mining

Felix Jungermann and Katharina Morik
TU Dortmund, Computer Science Department, LS VIII

February 21, 2008

Abstract

Where Information Retrieval (IR) and Text Categorization delivers a set of (ranked) documents according to a query, users of large document collections would rather like to receive answers. Question-answering from text has already been the goal of the Message Understanding Conferences. Since then, the task of text understanding has been reduced to several more tractable tasks, most prominently Named Entity Recognition (NER) and Relation Extraction. Now, pieces can be put together to form enhanced services added on an IR system.

In this paper, we present a framework which combines standard IR with machine learning and (pre-)processing for NER in order to extract events from a large document collection. Some questions can already be answered by particular events. Other questions require an analysis of a set of events. Hence, the extracted events become input to another machine learning process which delivers the final output to the user's question. Our case study is the public collection of minutes of plenary sessions of the German parliament and of petitions to the German parliament.

1 Introduction

Several information systems make available large collections of documents through the internet. Their documents can not only be retrieved by a search engine, but also by a built-in retrieval service based on the structuring of the

content. To the user, the making of the structures is hidden. The structure is presented in terms of categories among which the user might choose in order to navigate to the desired document. Although search for documents becomes more focused and user-driven, the user needs to understand the categories. Having a particular question in mind, the user is guessing under which heading she might find a relevant document. Moreover, the user needs to read the document in order to determine the answer to her question.

Some systems offer full-text search so that snippets of text are returned which include the keyword of the query. This eases already the burden of reading, but still the user needs to compose the answer out of some text excerpts. Again, the keyword needs to be chosen carefully in order to receive the right parts of relevant documents.

In contrast to the retrieval of documents, the Message Understanding Conferences (MUC) focussed on the extraction of structured information from natural language texts. Event extraction means to fill in the slots of a frame with named entities (NE) of the appropriate type, e.g., person, location, organisation or temporal and numeric expressions (cf. the more recent work [3]). Hence, NER became a subtask in its own right within MUC-6 and MUC-7 [1, 2]. Methods ranged from linguistic rules over pattern-based approaches to machine learning techniques. Linguistic knowledge is not only exploited by the hand-written rule or pattern-based extractions, but also when applying learning algorithms. For instance, part of speech (POS) tagging delivers class, case and number features of words, dictionaries classify known instances of NE types, and tagged texts allow to retrieve the context of NEs which becomes additional features to the word in focus. A knowledge-poor approach has applied machine learning to construct the required linguistic resources (e.g., name lists, gazetteers) in a bootstrap manner when learning NER [18]. Tagging documents with NER already offers some services to the user, namely highlighting words or phrases in the text. This might help the user to selectively read only the relevant parts of a long document.

A document is represented as a sequence of words. It is not a set of words, because the same word may occur at different places in the test, belonging to different tag-categories. For instance, the word "Paris" should be tagged PERSON in the sequence "Paris Hilton", but LOCATION in the sequence "stayed at the fashion week in Paris". The full sentence is represented in the following way: Paris(1,...,PERSON), Hilton(2,...,PERSON), stayed(3,...,O), at(4,...,O), the(5,...,O), fashion(6,...,O), week(7,...,O), in(8,...,O), Paris(9,...,LOCATION).

Definition 1.1 *Given a set of training examples of the form w(position, features, tag), find the function* $r : W \rightarrow T$*, where* W *is the set of word*

occurences and T is the set of tags including the O-tag.

Problems in NER are word sense disambiguation and the recognition of unknown words - words which never appeared in the training-dataset. Primal systems for NER were based on dictionaries containing many words or parts of words for labeling named entities. But especially these systems were not able to handle word sense disambiguation and the recognition of unknown words in a desirable manner. In both cases one has to examine the context of the currently processed word in addition to the word itself.

Systems based on machine learning techniques are hidden markov models, maximum entropy markov models, support vector machines, structural support vector machines and conditional random fields.

Currently, the restriction to NER is being dropped and approaches towards event extraction are undertaken, anew. Relation extraction aims at recognizing semantic relations between NEs, e.g., interactions of proteins [4]. Again, the hand-written rules were followed by learning approaches, first by learning the extraction rules [5]. Synactic knowledge was used by patterns for the extraction of relations [7] and syntactic dependency trees were used as features for probabilistic learning [8]. Learning approaches outperformed the hand-written ones. Relation learning removes irrelevant occurrences of NEs, selecting only the ones in the relation of interest. Hence, readers are confronted with a smaller number of text excerpts.

Event extraction is similar to relation extraction, but usually events contain more slots than relations have arguments. Several definitions are possible. Here, we simply define relations as parts of events and the relation learning in the way of [8].

Definition 1.2 *Given a set of documents D and an n-ary relation schema R with arguments $A_1, A_2, ..., A_n$, find instances $r(x_1, x_2, .., x_n)$ with $x_1 \in dom(A_1), x_2 \in dom(A_2), ..., x_n \in dom(A_n)$ in D.*

Typically, the relation r is represented in natural language by a verb, and the domains of arguments can be constrained by the case of a noun and a NE type. Typically, the arity is small, $n \leq 4$. We could write the form of events similarly, just allowing more complex arguments. Where a relation instance has just a (possibly composite) word as argument, an event may have a NE, a phrase, a relation, or a reference to another event as argument. In order not to confuse the reader, we use another notation for events.

Definition 1.3 *Given a set of documents D and a schema E defining slots $S_1, ..., S_n$ as elements, find instances $< e >< S_1 > ... < /S_1 >, < S_2 > ... < /S_2 >, ..., < S_n > ... < /S_n >< /e >$ in D.*

Typically, the schema corresponds to an XML schema, the slots correspond to XML elements, and instances are tagged text. Since XML elements can be structured themselves, relations can become slots of an event. The name of the schema corresponds typically to a noun, indicating the type of the event. Event extraction allows to build up a database. This, in turn, allows to do analyses ranging from simple queries over statistics to knowledge discovery (data mining). Although not being comparable to a natural language dialog system, more service is offered to the user.

In this paper, we propose to combine IR, NER, relation extraction, event extraction, and data mining in order to offer more services to users. We illustrate our framework by the application of the German Parliament document collection (see Section 2). The services we are aiming at are introduced in Section 3. The IR techniques and how we are using them for answering simples questions and as a basis for following steps are presented in Section 4.1. In Section 4.2 the named entity tasks are described, Section 4 describes the information extraction tool we are developing as a plugin to the data mining tool RapidMiner [15].

2 The German Parliament Application

The website of the German parliament (`http://www.bundestag.de`) is an excellent example of a web-based information system. It is structured according to the categories: parliament, members, committees, documents, knowledge, European Union, international, and visitors. To each category, a number of documents with links to other content is stored.

In particular, all plenary sessions are documented, from the 8th period until today (16th period). Also the printed papers which form the basis of discussions and decisions in plenary sessions, the recommendations of committees, small and large interpellations ("kleine Anfrage", "Anfrage"), legal proposals, are available as well as information about the members of parliament. We are focusing on the document collection, here. There is an information system, DIP21 (`http://dip21.bundestag.de`), which already offers some services to process the documents.

These services seem to merely use an index over the given documents. The documents are available in PDF format (mostly), the pages of the members of parliament are written in html. The identifier numbers of printed papers and plenary sessions need to be explained. For each plenary session there is an agenda, where to each of its points printed papers form the basis. In the plenary session, a topic is called by the numerical identifier of the respective printed paper. For instance, a committee might have decided to

recommend the rejection of a request. This means, that there is a printed paper with, let's say, ID=16/5540 which proposes something, i.e., is a *request*. The committee's *recommendation* to reject the proposal has another number, e.g., ID=16/5561. In the plenary session, the parliament decides about the recommendation 16/5561. If the recommendation is accepted, this means, that the request 16/5540 is rejected. If the recommendation is rejected, the request must again be discussed (and changed) in the committee yielding another paper of type *changed request* with a new number, e.g., ID=16/6102. For users who just want to know, for instance, whether the old age pension increases, or not, it is cumbersome to follow all these references through all the documents. Hence, for users it is not easy to actually receive answers to their information needs, although all information is publicly available.

Such a situation is quite common for web-based information systems. We use the one of the German parliament, because it is publicly available and we, as citizens, are allowed to analyse the documents. The goal is to enhance the services for users by moving more into the documents.

While the German Parliament web site is typical with respect to the services it offers, it is rather exceptional with respect to the language. Analysing word frequencies, we receive the typical power law distribution. However, the word length is even for German extraordinary. There are words like
"Konsensfindungserleichterungsmassnahme",
"Fernstrassenbauprivatfinanzierungsgesetzesänderungsgesetz", or
"Grundstücksverkehrsgenehmigungszuständigkeitsübertragungsverordnung".
Our corpus of the periods 13 - 16 contains 50,363 documents with about 470,000 different words. This shows that the political language is extremely challenging for extraction purposes.

3 Services

Services of the parliament's web-site are currently restricted. We would like to offer more to users. There is a variety of questions which people like to ask assuming different answers. For instance, the following questions were raised by a group of our students:

1. How many members of parliament have children?
2. How many of the female members of parliament have children?
3. Which requests (which plenary sessions) dealt with the relation between Germany and Turkey?
4. Which decisions were related to students from foreign countries?

5. Under which aspects have the relationship between Germany and Turkey been discussed?

6. Which party has signed the most requests?

7. Which party has the most requests rejected?

8. How many changes were necessary before the law for unemployed (Hartz law I, II, III, IV) has been decided?

9. Which parties or members of parliament were against the student registration fees?

10. Given the three levels of additional income, is there a correlation between the party and members with the highest level of additional income?

11. Which events were used as argument in favor of restricting civil rights in favor of enhancing the state's security?

The questions demand for different types of results.

- Some questions just ask for excerpts of documents, questions 3 and 4 are examples of this type. The answer set can be determined by full-text search and some post-processing (cf. Section 4.1).

- Some questions ask for counts of entities which are easy to recognize, questions 1 and 2 are examples of this type (cf. Section 4.1).

- Some questions ask for statistical analysis of relations, question 9 is an example of this type (cf. Section 4.2).

- Some questions ask for counts of events, questions 7, 8, 9 are examples (cf. Section 4.2).

- Some questions ask for statistical analysis of events or machine learning, questions 7 and 10 are examples (cf. Section 4.3).

- Some questions demand text understanding, questions 5 and 11 are examples. We exclude these difficult questions, here.

The architecture of our system for targeted information extraction is shown in Figure 1.

The graphical user interface offers menues for simple questions, questions about relations or events, and questions requiring some statistical analysis or

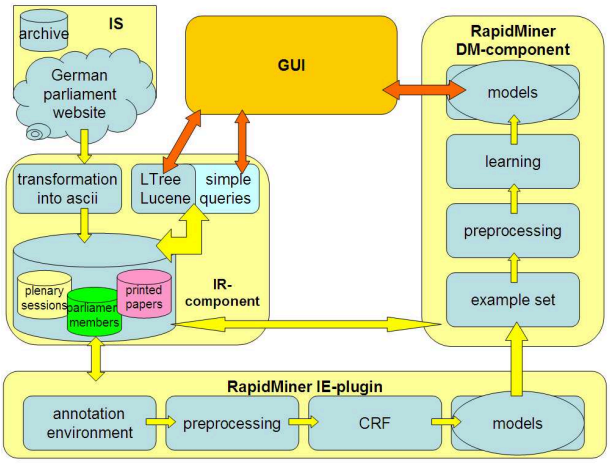

Figure 1: System design

learning. Documents from the web-based information system are retrieved, transformed from PDF into ascii format, and stored in the document base, which is then indexed. The simple questions are answered on the basis of the document's index. Some annotations are quite easy and can be achieved on the basis of simple patterns. The annotated documents become stored in the document base, as well. Questions about easy to recognize entities are answered on the basis of these data. Other annotations according to NE require to learn the NER. This is performed by the IE-plugin (see Section 4.2). Again, annotated documents are stored for further use in the document base. In order to learn about relations and events, the regular RapidMiner with designed experiments is integrated.

This architecture is quite general and a blueprint for a class of applications, namely enhancing web-based information systems. Depending on the application is the particular set of pre-processing operators, the regular expressions or other patterns to be used, and the particular sets of relation and event schemata. Due to RapidMiner's flexibility, it is easy to adapt the system to a new domain.

4 Targeted Information Retrieval

In this section we explain how we prepare for answering various questions (see Section 3) using our system or particular components of it.

Referring to Figure 1 one can see that our system consists of three components which are now presented in detail.

4.1 IR-Component

Many questions already are to be answered by IR techniques. To get these questions answered is on the one hand a nice benefit for users and on the other hand relatively easy to offer.

The system first of all extracts all the plenary session documents and printed papers and stores them for later use in ascii-format. Additionally the websites of the members of parliament are extracted in order to fill personal event templates which contain names, birthday, birthplace, family status, children, education and so on. In addition we – as written in Section 2 – build up an index of all the documents using the open-source indexing environment *lucene* (`http://lucene.apache.org`).

Furthermore, we extract the information of every printed paper – similar to the approach using for the members of parliament – into an event-like template for accessing the information easily. This extraction by now is done using trigger-words like, e.g., "geboren in" (born in) for birthplaces.

The printed papers follow a special formatting which helps to extract the number of the printed paper, the members of parliament, the parties involved, the date, and an abstract of the printed paper. These are the slots of the printed-papers-template.

The member-of-parliament-templates, the printed-paper-templates, the original (complete) documents and the index over all documents form our repository which is ready to use for answering questions 1, 2, 3, 4 and 6. The data of the repository is used for further analysis in the IE-component and in the Data Mining(DM)-component in order to answer more difficult questions.

Our system contains all printed papers of the 14th, 15th and 16th period as templates currently. Requests like *'How many requests has the party SPD signed?'* or *'Show me all the printed papers of type "Gesetzentwurf"!'* can be easily processed.

4.1.1 Experiments using the IR-component

As an example which will be picked up again in Section 4.3 we extracted all the requests and its corresponding recommendations of the committees either to reject, to accept or to depose the proposals. For this task, one has to look for all the printed papers of type *request*. Then, one has to extract all the *recommendations* considering these requests. Finally, one has to search snippets according to the request numbers in the recommendations to find the decision of the committees. Users now can easily look at relevant excerpts of large documents, focused on particular recommendations to see

what happened to a request.

The recommendation is a relation embedded into the event *request* consisting of the outcome (i.e., accept, reject, depose), the number of the recommendation and the recommending committee (e.g. committee of justice). The following, for instance, is a positive recommendation of the committee of justice (in German: 'Rechtsausschuss'):
$recommend(< printed\_paper > 14/358 < /printed\_paper >, < recommendation > accept < /recommendation >, < committee > Rechtsausschuss < /committee >$
)

4.2 IE-Component

The IE-component is profitable for users when, for example, highlighting special objects (NEs) in the texts achieving a better user-guidance. It is necessary for other processes for which it captures trigger-words or patterns. Additionally, the IE-plugin delivers the relation extraction (see 4.1.1).

As learning and information extraction environment we use the open-source software RapidMiner[1] [15] which is implemented in Java. RapidMiner is first of all a well-suited and worldwide used data-mining tool which allows to design complex chains of operators. Pre-processing, learning, and evaluation operators are implemented and can be combined to form a so-called experiment (nested operator chain). RapidMiner offers several interfaces to other systems and an elegant mechanism to plugin specialized add-ons. We have developed such a plugin for information extraction (IE). RapidMiner represents examples as a set of attribute-values together with a label indicating the class membership. In case of data-mining or machine learning, the individual examples do not affect each other. In contrast, natural language processing (named entity recognition, for instance) has a sequential character which should not be destroyed. Keeping that in mind, one has to maintain the sequential character on the one hand and the clustering-character of sentences on the other hand, if one wants to express natural language utterances as examples in RapidMiner. Converting natural language to examples means that every word occurrence becomes one example which has a bunch of attributes. Preceeding and following words become attributes of the current word. Corresponding to the learning task (in this case NER), a label attribute is added (the NE).

Using the IE-plugin for RapidMiner one must first of all define a dataset (text). Then, one can use various pre-processing-operators. Corresponding to McDonald's definition of internal and external evidence ([14]) the categorization of words depends on internal features – extracted directly from the form of a word – and external features features – extracted from the context of a word. The pre-processing operators make use either of internal or of external evidence. Considering current

[1] formerly known as YALE

work on sequence labeling ([12]) one sees that there is a set of features which delivers good results. These features consist of character n-grams, prefixes, suffixes, word-generalizations and so on. The external evidence is used by encoding the knowledge of surrounding contexts into attributes of words. So particular words also have attributes corresponding to words in front or behind the word. Most of the features presented in [12] also are implemented in our IE-plugin.

After the pre-processing steps, one can use multiple machine learning methods. In case of sequence labeling tasks, [16] showed that the structured support vector machine (SVMstruct [19]) delivers the best results compared to other methods like hidden markov models and especially conditional random fields (CRF) ([11] and the following subsection). But [9] showed that the better performance of the SVMstruct compared to CRFs is only due to different internal features used for learning. Their comparison showed that the perfomance of both methods are nearly the same. Hence, both methods can well be used. Since the SVMstruct is not yet implemented in Java, we integrated CRFs into RapidMiner – we use MALLET [13] as basis for our CRF-operator.

The plugin also contains an annotation-functionality for annotating texts by hand or correcting wrongly annotated texts in order to prepare a training set.

4.2.1 Conditional Random Fields

Conditional random fields (CRF) are based on Markovian random fields (MRF). Markovian random fields are undirected graphical models which consist of an acyclic graph containing vertices and edges. The vertices accord to states. The states may have notations called label. The most probable configuration of the states - labeling - is calculated over a set of potential functions. Because the graph is undirected, particular states do not have any preceding state but one or more neighbouring states they are connected with. In MRFs the Markovian property holds, meaning that the probability for a special configuration of a state just depends of the neighbouring states (vertices) (see formula 1).

$$P(q_v|q_w : v \neq w) = P(q_v|q_w : v \sim w) \tag{1}$$

All vertices in a graph connected to each other build a clique. The potential functions for MRFs are therefore defined on cliques and the amount of potential functions is defined as:

$$\Phi = \phi_{c_i} : A_{c_i} \rightarrow \mathbb{R}^+ \tag{2}$$

Formula 2 shows that every labeling of a clique leads to a positive, real-valued output (which is not restricted to be between 0 and 1).

The probability for a special labeling $\vec{y}$ is defined as:

$$P(\vec{y}) = \frac{1}{Z} \prod_{\Phi} \phi_{c_i} \left(y_{i_1} ... y_{i_{c_i}} \right) \tag{3}$$

Table 1: Recommendation extraction for all requests

Found requests	1.935
Requests without recommendation	680
Requests with recommendation	1.255
Recommendation 'reject'	794
Recommendation 'accept'	251
Recommendation 'depose'	44
Recommendation not extractable	166

and Z is a normalization factor converting the output of formula 2 into a probability (between 0 and 1).

$$Z = \sum_{\vec{y} \in Y} \left[\prod_{\Phi} \phi_{c_i} \left(y_{i_1} ... y_{i_{c_i}} \right) \right] \tag{4}$$

One could say the probability for a given label-sequence is the output of all the potential functions for this label-sequence divided by the output of all potential functions for every possible label-sequence.

In contrast to MRFs, CRFs are conditional on an observation sequence $\vec{x}$. Given an amount of potential functions the goal of the training-phase is to set up the weights λ for maximizing the log-likelihood (formula 6). The probability for a label-sequence, given a special observation-sequence and the weight-vector $\vec{\lambda}$, is calculated as seen in formula 5.

$$P(\vec{y}|\vec{x}, \vec{\lambda}) = \frac{1}{Z(\vec{x}, \vec{\lambda})} \exp \left[\sum_{k} \sum_{i=1}^{t} \lambda_k \phi_k (y_{i-1}, y_i, \vec{x}, i) \right] \tag{5}$$

$$L(\vec{\lambda}) = \sum_{h \in E} log P(y^{(\vec{h})} | x^{(\vec{h})}, \vec{\lambda}) \tag{6}$$

In test-phase the most probable label-sequence for a given observation-sequence is calculated efficiently using the viterbi-algorithm.

4.2.2 Relation Experiment using the IE-plugin

On the basis of the snippets according to the request numbers in the recommendations, we have extracted the relation *recommend* with the arguments *reject*, *accept*, and *depose*. We present the results in Table 1.

4.2.3 NER Experiments using the IE-plugin

We made an exemplary NER-experiment using a manually annotated document of the plenary sessions. The document contained about 2.700 sentences containing

Table 2: NEs in the examined document (printed papers (p.p.), plenary session (p.s.))

	name	institution	party	person	location
count	1.115	823	703	422	409
f-measure in %	90,3	87,9	96,1	68,4	68,2

	organisation	reaction	date	p.p. no.	p.s. no.
count	238	120	78	45	18
f-measure in %	52,1	55,4	60,3	37,7	36,7

nearly 56.000 words or in other words 'tokens' – points for example are tokens, too. Table 4.2.3 shows the numbers of NEs and the performance (measured with *f-measure*) which was achieved on the dataset using a ten-fold-cross-validation. The results show that some NEs are easy to spot like, for instance, name and party. In contrast, the recognition of a plenary session and of printed paper numbers is difficult for NER. However, the numbers and the dates do not need to be extracted by NER, but can be easily set by regular expressions during pre- or post-processing. True NER tasks are recognizing institutions, locations, and organizations. Without background knowledge, organisations are hard to detect.

4.3 DM-Component

The innovation of our system is the opportunity to use extracted events as input for data mining experiments. It is nice to get questions answered like *'How many requests are recommended to be rejected?'*, but data mining goes beyond that. It offers the opportunity to get to know why or under which circumstances a request has been rejected. We are using RapidMiner for data mining. According to the example of Section 4.1.1 we converted all found requests into examples as an input for a data mining task. The event *request* has the following form:

```
<e type = "request">
  <printed_paper>14/138</printed_paper>
  <recommend>
    <printed_paper> 14/358 </printed_paper>
    <recommendation> accept </recommendation>
    <committee> Rechtsausschuss </committee>
  </recommend>
  <party 1> SPD </party 1> <party 2> BUENDNIS90/DIE GRUENEN </party 2>
  <party 3> null </party 3> <party 4> null </party 4>
  <party 5> null </party 5> <multiMOP> false </multiMOP>
  <justParty> true </justParty> <government> false </government>
</e>
```

There are only 5 parties in the parliament. The supporters of a request can be a number of members of parliament (MOP), some parties, or the complete government. The slot $< party1 >$ indicates the party initiating the request. The shown example states that two parties together have signed the request 14/138 and the request got a positive recommendation by the committee of justice. This event is transformed into a data set for learning, where the slots become attributes. The recommend arguments for the decision are encoded by numbers: 0 for no recommendation, 1 for accept, 2 for reject, and 3 for depose. These arguments are the class labels for learning. A simple decision tree learner delivers following results.

Decision-Tree-Experiment on extracted events

```
Tree

PARTY 2 = CDU/CSU
|   PARTY 1 = SPD: 0 {0=37, 2=0, 1=4, 3=0}
|   PARTY 1 = PDS: 2 {0=0, 2=2, 1=0, 3=0}
PARTY 2 = BUENDNIS 90/DIE GRUENEN
|   Just Parties = true: 0 {0=95, 2=1, 1=44, 3=3}
|   Just Parties = false: 1 {0=75, 2=1, 1=155, 3=3}
PARTY 2 = null
|   PARTY 1 = SPD: 2 {0=44, 2=164, 1=1, 3=9}
|   PARTY 1 = PDS: 2 {0=171, 2=413, 1=9, 3=20}
|   PARTY 1 = CDU/CSU: 2 {0=46, 2=108, 1=1, 3=3}
|   PARTY 1 = BUENDNIS 90/DIE GRUENEN: 0 {0=141, 2=102, 1=2, 3=6}
|   PARTY 1 = CDU: 2 {0=0, 2=1, 1=0, 3=0}
|   PARTY 1 = FDP: 0 {0=2, 2=0, 1=0, 3=0}
|   PARTY 1 = DIE LINKE.: 0 {0=1, 2=0, 1=0, 3=0}
PARTY 2 = SPD
|   Just Parties = true: 0 {0=22, 2=0, 1=3, 3=0}
|   Just Parties = false: 0 {0=31, 2=0, 1=23, 3=0}
PARTY 2 = DIE LINKE.: 2 {0=0, 2=1, 1=0, 3=0}
```

The possible label allocations are given in brackets at the leaves of the learned tree. Using this little number of attributes one would not think of getting interesting results, but there are some: if the attribute *PARTY 2* is null (just one party is signing the request), one can see at the leaf with *PARTY 1 = PDS* that most of the requests of this party, the leftist party, are recommended to be rejected. This answers question 7 from the Section 3. A ten-fold-cross-validation over the requests ended up in *67,1 % accuracy* to predict the label.

5 Related Research and Conclusion

The first trainable systems for event extraction were based on wrapper induction (WI) [10]. WI-based systems are processing a huge amount of already structured data – typically labeled by html-tags. By learning which tags *wrap* the interesting data, WI-based systems are capable of filling event slots after having found special tags. These systems work well if and because the html-syntax is well-formed and thus offers kinds of slots, already.

A more specific problem is the analysis of semantic roles and its corresponding relations. [20], for instance, tried to extract patterns for the relation 'position statement' automatically – given a little seed example set. Their results show that this automatic pattern extraction works as good as hand-crafted ones, being not as time-consuming. Actually 'relational search' is used to extract relations between entities automatically without using seed-examples [6] in an unsupervised and highly scalable manner. Therefore any object-string-pair which occurs in a neighbouring context in a (huge) document collection is extracted with its corresponding relation – which just is the text between the two object-strings. An *extraction graph* – consisting out of objects and relations between these objects – is built up in order to answer queries which cannot be answered easily by traditional search-engines. [17] present a similar system which uses an ontology to annotate semantic categories (NEs) in texts and in turn uses the annotated texts to update and advance its ontology.

Our plugin for NER is ready for use and its performance is competitive with other approaches. Enhancements of performance can be achieved by adding background knowledge in terms of lists of organizations and locations. Further experiments will be made with other NEs. Possibly, these demand for further feature extractions.

Related systems like the one from [17] or the one from [6] are powerful for relation extraction. It should be investigated if these techniques are applicable for the event- and relation-extraction in our system, and whether it would improve our extraction of events.

Our relation extraction is currently quite heuristic. Here, more sophisticated approaches could extract relations from the plenary sessions. It is extremely hard to extract the opinion (in favor, against) from speeches in parliament. The language used is most elaborated and full of subtle irony. Currently, we restrict ourselves to the ballot results where the language is more standardized.

Similarly, event extraction from the plenary sessions are currently referring to decision making events (recommendations, decisions, votes, passes of a law), interjections, and the extraction of the list of speakers in a plenary session dealing with a certain topic. This is already challenging, since the speeches can be nested, and the topics are called in various ways. Far more difficult is to recognize the position in a speech. For instance, does a speaker argue in favour or against nuclear plants? Extracting the opinion of a politician at several points in time would allow

to register changes in the political belief of a member of parliament. However, the difficulty is the extraction of the opinion from text. Hence, we work on events which relate requests (standing for a political position) and politicians or parties. The summarizing text of a request represents the position and is only interpreted by the reader.

The major focus of this paper is the combination of IR, IE, and DM. Indexing services and information extraction transform a document collection into a set of events and relations which form the basis of data mining. The efforts of finding relevant paragraphs in the documents, of writing regular expressions for the extraction of simpler entrities such as, e.g., document numbers or dates, are turned into operators of the IE plugin of RapidMiner and, hence, are available for other applications, as well. The IE-plugin offers these preprocessing operators. It interacts with the IR-tool *lucene*. It offers an annotation tool and runs the CRF NER in a loop of cross validation. Hence, we have not merely shown an application but the development of a system which eases to build an application. Such a principled approach to enhancing services of web-based information systems by event extraction and data mining is a novelty, as far as we know.

The human-computer interface is currently under development. It will accept queries and call the respective IR-, IE-, or DM-processes. However, the major issue for a user-friendly HCI is to prepare the answers. We have shown that calling the processes manually already delivers the relevant text snippets, the tagging of some NEs, and the decision trees, respectively. Experiments using the IR- and IE-component deliver good results which can be used as input for the DM-component. First experiments in 4.3, using data mining techniques for analyzing extracted templates which represent specific relations show interesting results.

6 Acknowledgments

This work was supported by the *Deutsche Forschungsgemeinschaft (DFG)* within the *Collaborative Research Center "Reduction of Complexity for Multivariate Data Structures"*.

References

[1] *Proceedings of the Sixth Message Understanding Conference*, Columbia, Maryland, 1995. Morgan Kaufmann.

[2] *Proceedings of the Seventh Message Understanding Conference*, Fairfax, Virginia, 1998. Morgan Kaufmann.

[3] Chinatsu Aone and Mila Ramos-Santacruz. REES: a large-scale relation and event extraction system. In *Proceedings of the sixth conference on Applied*

natural language processing, pages 76–83, Seattle, Washington, 2000. Morgan Kaufmann Publishers Inc.

[4] Christian Blaschke and Alfonso Valencia. Can bibliographic pointers for known biological data be found automatically? protein interactions as a case study. *Comparative and Functional Genomics*, 2:196–206, 2001.

[5] Razvan Bunescu, Ge Ruifang, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk Wah Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Journal of Artificial Intelligence in Medicine*, 2004.

[6] Michael J. Cafarella, Michele Banko, and Oren Etzioni. Relational web search. Technical report, University of Washington, CSE, 2006.

[7] Udo Hahn and Martin Romacker. An integrated model of semantic and conceptual interpretation from dependency structures. In *Proceedings of the 18th conference on Computational linguistics*, volume 1, pages 271–277. Association for Computational Linguistics, 2000.

[8] Sophia Katrenko and Pieter Adriaans. Learning relations from biomedical corpora using dependency trees. *Knowledge Discovery and Emergent Complexity in Bioinformatics*, Volume 4366/2007:61–80, 2007.

[9] S. Sathiya Keerthi and S. Sundararajan. Crf versus svm-struct for sequence labeling. Technical report, Yahoo! Research, 2007.

[10] Nicholas Kushmerick, Daniel S. Weld, and Robert Doorenbos. Wrapper induction for information extraction. In *Proceedings of IJCAI-97*, 1997.

[11] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[12] Robert Leaman and Graciela Gonzalez. Banner: An executable survey of advances in biomedical named entity recognition. In *Proceedings of the Pacific Symposium on Biocomputing 13*, pages 652–663, 2008.

[13] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

[14] D.D. McDonald. Internal and external evidence in the identification and semantic categorization of proper names. In B. Boguraev and J. Pustejovsky, editors, *Corpus Processing for Lexical Acquisition*, pages 21–39. MIT Press, Cambridge, MA, 1996.

[15] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. YALE: Rapid Prototyping for Complex Data Mining Tasks. In Tina Eliassi-Rad, Lyle H. Ungar, Mark Craven, and Dimitrios Gunopulos, editors, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pages 935–940, New York, USA, 2006. ACM Press.

[16] Nam Nguyen and Yunsong Guo. Comparisons of sequence labeling algorithms and extensions. In *In Proceedings of the International Conference on Machine Learning (ICML 07)*, pages 681–688, June 2007.

[17] Borislav Popov, Atanas Kiryakov, Dimitar Manov, Angel Kirilov, Damyan Ognyanoff, and Miroslav Goranov. Towards semantic web information extraction. In *In Proceedings of the ISWC'03 Workshop on Human Language Technology for the Semantic Web and Web Services*, pages 1–21, 2003.

[18] Marc Roessler and Katharina Morik. Using unlabeled texts for named-entity recognition. In Tobias Scheffer and Stefan Ruping, editors, *ICML Workshop on Multiple View Learning*, 2005.

[19] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[20] Roman Yangarber and Ralph Grishman. Machine learning of extraction patterns from unannotated corpora: Position statement.